The Architecture of a Resilience Infrastructure for Computing and Communication Systems

Algirdas Avižienis
University of California, Los Angeles, USA, and
Vytautas Magnus University, Kaunas, Lithuania

Presented to IFIP WG 10.4, Visegrad, Hungary, June 29, 2013

Please Note a Contemporary Paradox:

Computer systems provide protective infrastructures
for critical infrastructures of modern society:
electrical power, telecommunications, transportation, …..
but:
these computer systems do not possess
a protective infrastructure of their own!

My Vision of a Solution:

To design a *generic*, *fault-tolerant*, *software-free Resilience Infrastructure RI* for computing and communication systems that should allow a significant simplification of the other defenses

The Deficiencies of Current Defenses

1. There are unprotected “hard core” elements, especially in the error detection and recovery management hardware and software
2. Hardware and software defenses are interdependent, thus both have to succeed in order to complete recovery

An Example of “hard core” and Interdependence

Pentium and Itanium processors have a Machine Check Architecture (MCA) in which hardware errors are recorded by setting bits in a set of MCA registers that are **not** protected by any form of redundancy or fault tolerance
The operating system then senses the MCA register bits and initiates recovery action

The Definition of Resilience

- **Resilience** is the persistence of dependability when facing changes *[J.-C.Laprie,"From Dependability to Resilience" Proc.of DSN 2008, Supplemental Volume, Fast Abstracts]*

  Changes are classified: (1) by nature,(2) by prospect,(3) by timing.

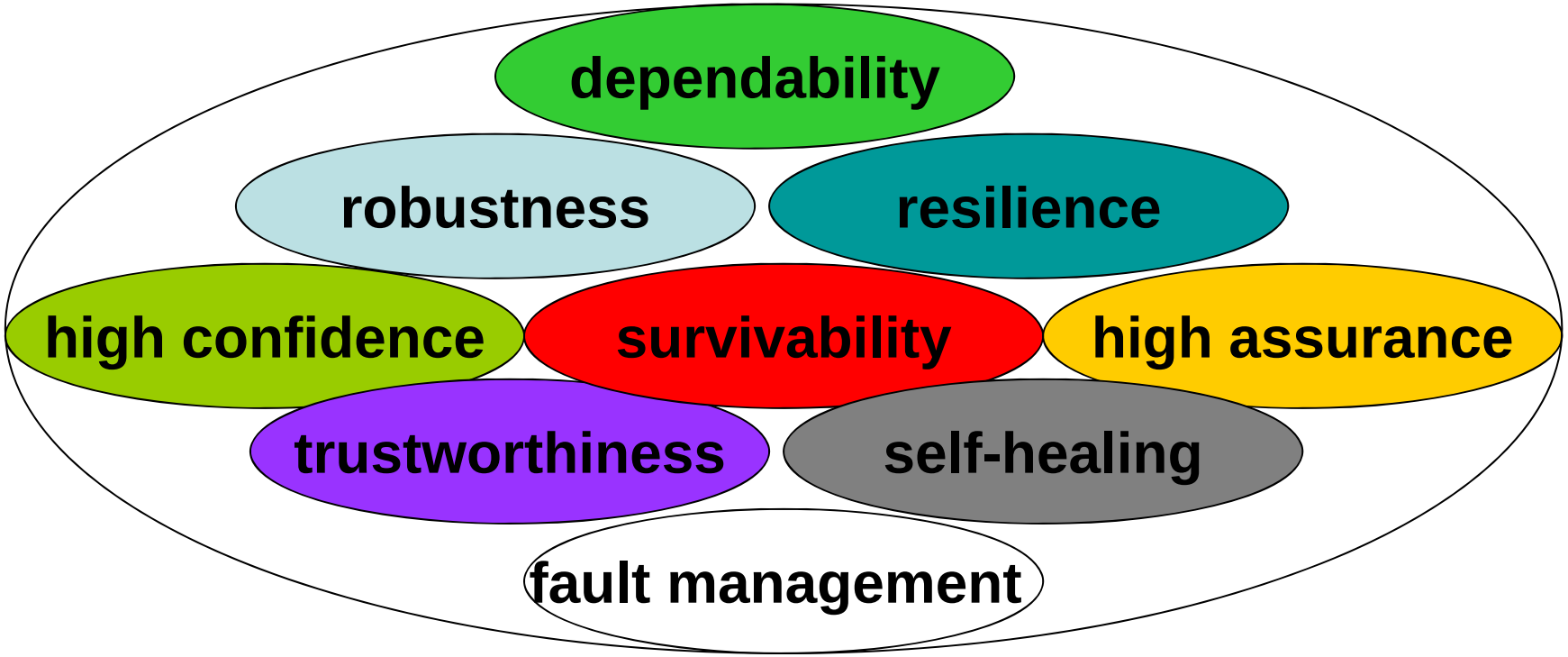
  My elaboration of the definition by Jean-Claude Laprie:

- **Resilience** is the ability of a system to sustain dependable operation in the presence of harmful changes that:

  (1) exceed the limits of expected threats, or
  (2) are not identified at all as expected threats in the system's dependability specification.

Our Field's Objective: deliver expected service under adverse circumstances

Our Field's Top Concepts:



How are they related ?

How Is Resilience Created ?

(1) System designers exceed the requirements of the dependability specification: either **(a)** inadvertently, or **(b)** deliberately. This is **Implicit Resilience**.

(2) System designers add new features to system architecture that are intended to provide resilience. This is **Explicit Resilience**

This presentation describes approach **(2):** a subsystem called the **Resilience Infrastructure** that can be attached to a **"Client"** system to provide resilience.

Attributes of the Resilience Infrastructure RI

(1) RI is functionally and physically **separate** from Client. The only links are status messages and requests from Client, and commands and request replies from RI.

(2) RI is **generic**: it can serve any Client system that can send status messages and requests, and receive commands and replies from the RI.

(3) RI is implemented by **hardware** and **firmware** only. Absence of software eliminates potential vulnerabilities

(4) RI is **fully self-protecting** by hardware fault tolerance techniques, including shutdown and restart sequences for catastrophic faults affecting the RI and Client.

Installation of the RI

(1) Given that the Client is composed of **N** subsystems called **C-nodes** that **(a)** have an On-Off switch for power, **(b)** form an error-containment region, and **(c)** can send messages and receive replies from the Resilience Infrastructure..

(2) The principal subsystem of the RI is the **Monitor** (**M-node**). One M-node is connected to an Affinity Group of **C-nodes** until each one of the N C-nodes has one M-node attached to it.

(3) Each C-node can send **(a)** status, **(b)** error, and **(c)** inquiry messages to its M-node.

(4) The M-node can respond to C-node messages by **(a)** response, **(b)** recovery, and **(c)** power on-off messages.

Components (“nodes”) of the RI

(1) The **M-node** contains a **read-only memory** that is pre-loaded by the Client’s designers with the response to every message from the C-node. It also contains non-volatile **status registers**, sequencing logic and hardware for fault tolerance. The fault-tolerant array of M-nodes is called an **M-cluster**.

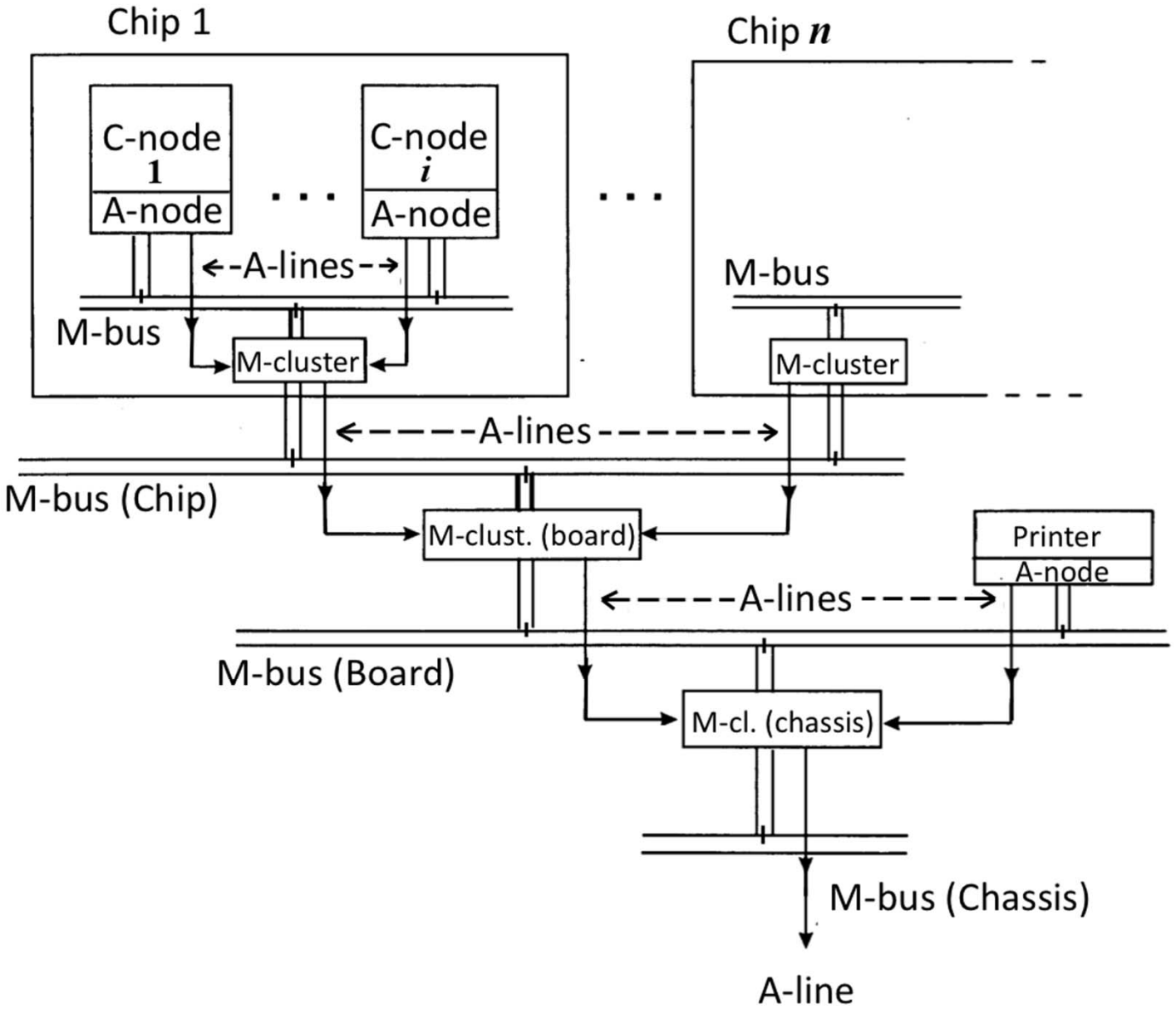
(2) The **A-node** (“Adapter”) is an auxiliary node that connects the C-node to its M-node in a fault-tolerant manner, as a self-checking pair. It also controls the power switch of the C-node according to commands from the M-cluster..

Components (“nodes”) of the RI (cont.)

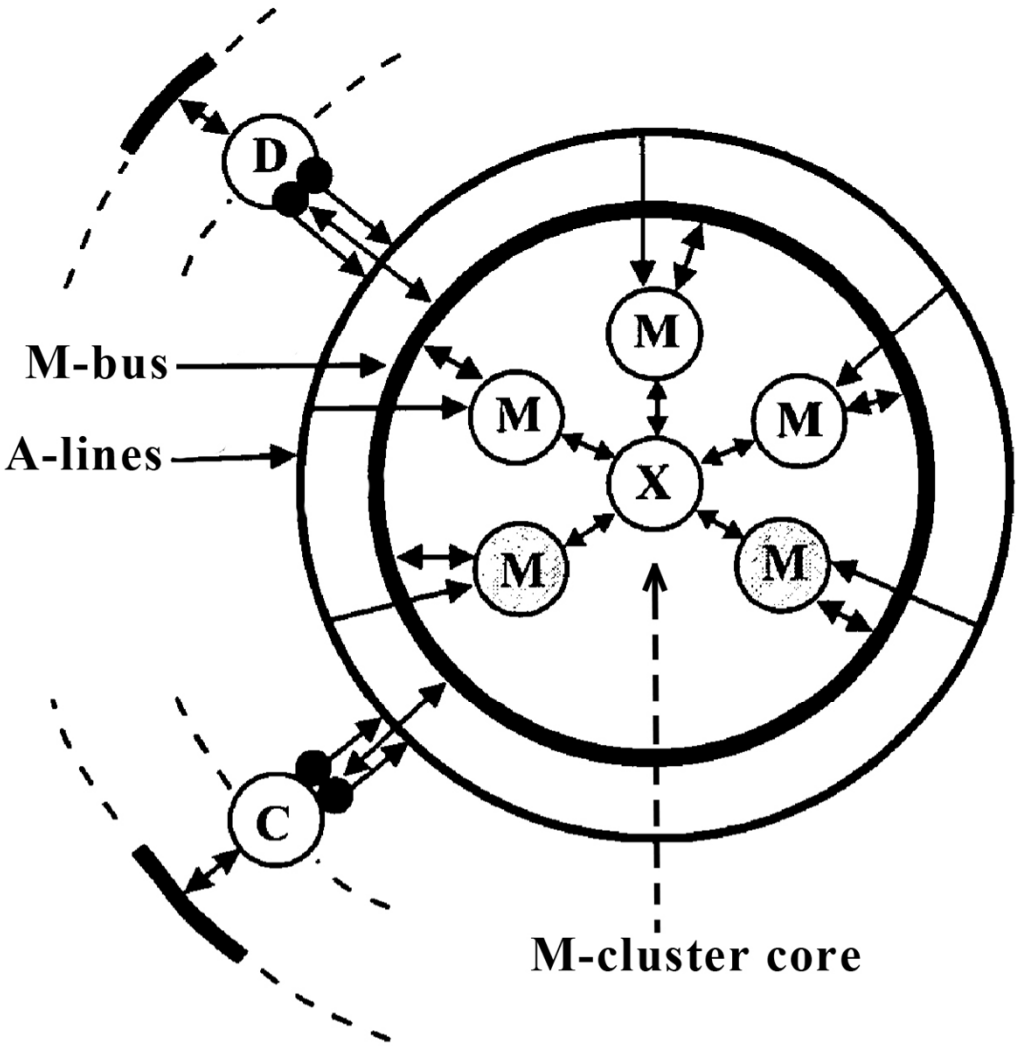
(3) The **S3** (startup-shutdown-survival) **node** is the “hard core” node of the RI. It supervises the M-cluster and executes **shutdown and restart** of the RI+Client system in the case of catastrophic faults.

The S3 node maintains **system time and configuration** in radiation hard, nonvolatile registers. The S3-node needs fault tolerance protection by a separate power supply and the most effective “multiple pairs of self-checking pairs" method.

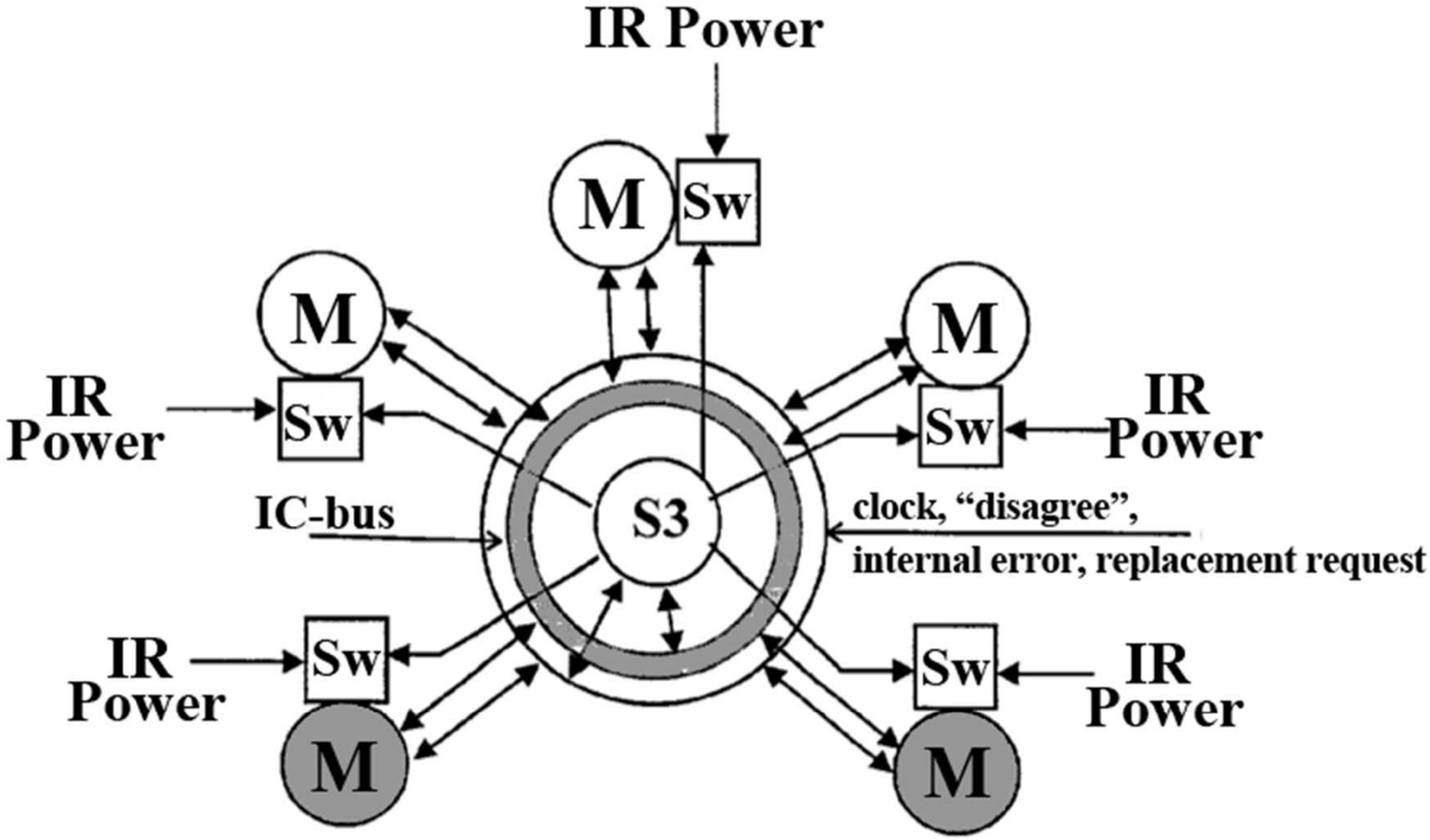
Hierarchical Structure of the RI



The M-Cluster



The M-Cluster M*



Will the Resilience Infrastructure Be Used ?

- I believe that the RI offers significant advantages, compared to the current implementations of fault tolerance, BIST, software monitors, and other defenses. The absence of software removes the source of many problems, and the complexity of the RI hardware is modest.
- However, there is a huge “legacy” problem – system design is a gradual process that cannot readily accommodate a big change – the transition to the inclusion of the resilience infrastructure.
- For the above reason I have identified the human exploration of **Mars** as a project that is sufficiently far in the future and also life-critical. For this reason the use of the RI could be considered in the initial definition of the system.
- It is important to note that any fault tolerance or other protective means of that future system can be accommodated – the RI is an additional feature that will **guard those guardians** when a catastrophic event threatens with disaster, although the RI can handle simpler dangers as well

The Manned Mission to Mars

- Important studies have been done since 2000:

  D. Landau and N.J. Strange "This Way to Mars", *Scientific American,* 306(6): 58-65 (December 2011).

  "Special report: Sending astronauts to Mars", *Scientific American,* 282(3): 40-63 (March 2000).

- My concern: the survival of the spacecraft for the 1000-day manned mission to Mars
- Assume: spacecraft systems are controlled by embedded computers that have state-of-the-art fault tolerance
- Remaining need: Assure resilience of the systems:
  provide defenses against unexpected, possibly catastrophic faults and make the defense mechanisms self-protecting
- My solution: Provide a hardware-based, fault-tolerant resilience infrastructure (RI) for all spacecraft systems

References

[1] Laprie, J.-C., “From Dependability to Resilience”, *Proceedings of DSN 2008, Supplemental Volume, Fast Abstracts, Anchorage, AL, June 2008.*

[2] Avižienis, A., Laprie J.-C., Randell, B., Landwehr C., “Basic Concepts and Taxonomy of Dependable and Secure Computing”, *IEEE Trans. on Dependable and Secure Computing, vol. 1, no. 1, 2004, pp. 11-33.*

[3] Avizienis, A., “Hierarchical Configurations in Error-Correcting Computer Systems “, *United States Patent No. US 7,861,106 B2, December 28, 2010*

[4] Avizienis, A., “Self-Testing and –Repairing Fault Tolerance Infrastructure for Computer Systems”, *United States Patent No. US 7,908,520 B2, March 15, 2011.*